

01 MARS 2019



PFE : THE INDEPENDENT SUPERVISOR

MANUEL DE MAINTENEUR | DII -2018/2019

BRANDON SIMON-VERMOT
ENCADRANT : M. AMEUR SOUKHAL

Commentaire	Date de modification
Création du document	27/02/2019
Ajout de l'introduction	27/02/2019
Ajout du contenu	01/03/2019

Table des matières

Introduction	3
Prérequis	4
NodeJS.....	4
Angular CLI.....	4
MongoDB.....	5
Robo 3T	6
Architecture de la Base De Données	7
Comment installer une dépendance ?.....	8
Comment accéder aux données ?	9
Comment éditer les données ?	10
Comment passer un utilisateur en administrateur ?	11
Comment éditer les scripts d'une application ?	12
Comment déployer l'application Web ?	12

Introduction

Dans le cadre de notre formation en école d'ingénieur, à Polytech'Tours, nous devons réaliser un projet de fin d'étude. Pour cela plusieurs sujets nous sont proposés, projets proposés par les différents professeurs de l'école ou proposé par une entreprise. Cependant il est aussi possible de proposer un projet personnel, ce qui est mon cas.

Pour proposer un projet, il faut au préalable fournir un cahier des charges et que le sujet soit étudié puis validé par un jury, tout cela se passe avant le début de l'année scolaire. Ainsi le sujet est évalué afin de déterminer s'il répond aux critères attendus.

Le contenu de ce manuel concerne les mainteneurs de l'application Web, les personnes qui mettront en place le site Web et toutes ces dépendances. Ce qui permettra d'obtenir une application Web fonctionnelle.

Prérequis

NodeJS

Au cours de ce projet, la partie Back-End est représenté par l'utilisation de NodeJS. Pour télécharger la dernière version, sachant qu'on utilise un système d'exploitation Linux, dirigez-vous directement sur le [site officiel](#) de NodeJS.

Démarrer le serveur NodeJS peut correspondre seulement à la partie Back-End mais aussi il est possible de regrouper la partie Front-End et la partie Back-End en une application, ce qui a été utilisé au cours de ce projet, de cette manière cela nécessite qu'un seul port. On utilise la commande suivante pour démarrer le serveur :

```
> npm run start:server
```

On peut retrouver cette commande dans le fichier de configuration « package.json ».

Angular CLI

La partie Front-End sera couverte par l'utilisation d'Angular, ne pas confondre avec AngularJS, ici nous utilisons la version utilisant du TypeScript. Pour télécharger Angular CLI, il est possible de passer par le gestionnaire de paquets de NodeJS, npm.

```
> npm install -g @angular/cli
```

Il est possible de démarrer indépendamment la partie Front-End via la commande :

```
> ng serve
```

MongoDB

L'installation de la Base De Données pourrait se faire via le cloud et configurer tout via cette solution, mais au cours de ce projet, l'installation directement sur le serveur dédié est privilégié, il est donc possible de télécharger « MongoDB Community Server », via le [site officiel](#), dans le but l'installer directement sur notre serveur et ainsi garder la main dessus.

Dans notre cas, l'installation du paquet debian permettra d'obtenir notre Base De Données directement sur notre serveur. Une fois installée, il est possible de lancer MongoDB comme un service via la commande :

```
> sudo service mongod start
```

Pour vérifier que le service est bien lancé, il est possible de consulter les logs disponibles via le fichier « /var/log/mongodb/mongod.log ». A l'intérieur de ces logs, il est possible de voir dans un fonctionnement normal, cette ligne :

```
[initandlisten] waiting for connections on port 27017
```

Elle indique que le serveur MongoDB attend des connexions sur le port par défaut 27017. Dans ce même fichier, il est possible d'y consulter tous les avertissements non critiques.

Pour redémarrer le service, il est possible de lancer cette commande :

```
> sudo service mongod restart
```

Ou pour arrêter complètement le service :

```
> sudo service mongod stop
```

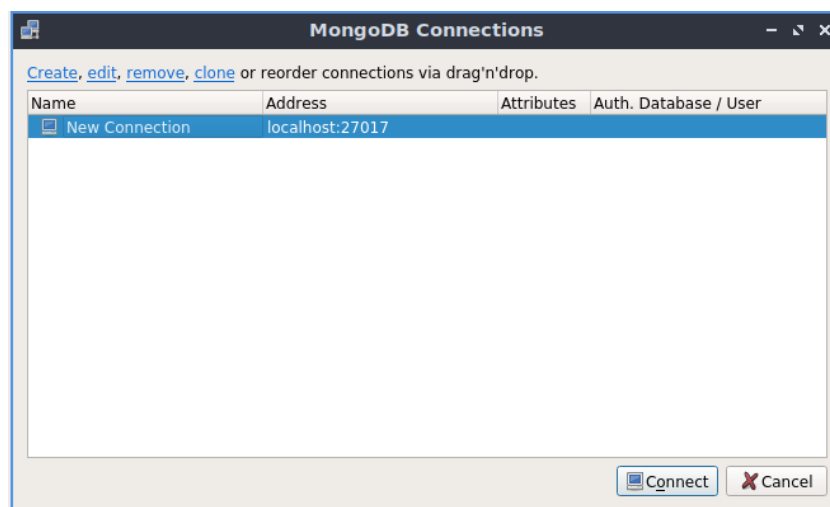
D'autres informations autour de MongoDB sont expliquées via les différents guides, disponible sur le [site officiel](#).

Robo 3T

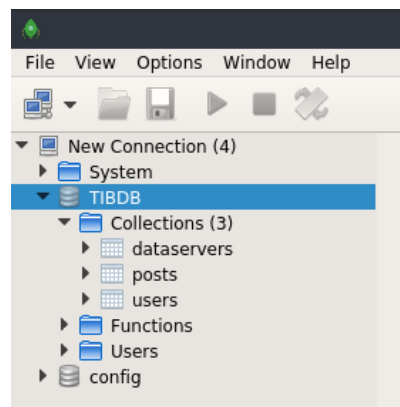
Robo 3T est une alternative gratuite à Studio 3T, réalisé par le même studio. Ces deux outils permettent d'accéder à notre serveur MongoDB via une interface graphique, il est tout à fait possible de s'en passer pour consulter des données de notre base de données, en passant par un terminal.

Afin de rendre la consultation plus facile et plus visuelle, l'utilisation d'un logiciel tel que Robo 3T regroupe ces avantages. Il est disponible en téléchargement sur le [site officiel](#).

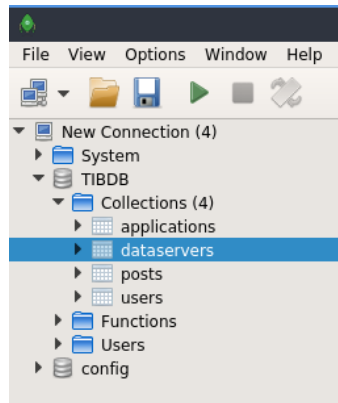
Une fois installé, au premier lancement Robo 3T vous propose de créer une connexion, via le lien « create », avec votre serveur de base de données.



Après s'être connecté à notre serveur, il est possible de voir sur la partie droite de l'application, toutes les bases de données qui constitue notre serveur, dans le cas du projet, la base « TIBDB » sera utilisé, ainsi on aura accès à nos compte utilisateur pour sécuriser notre base de données, mais aussi toutes les collections qui compose notre base.



Architecture de la Base De Données



A ce stade du développement du projet, la base de données est composée de quatre collections, permettant de stocker les différentes informations liées à nos suggestions, nos applications, les différents utilisateurs mais aussi les données liées à notre serveur dédié.

applications	
PK	<u>uniqueId</u>
	title
	content
	imagePath
	date

posts	
PK	<u>uniqueId</u>
	title
	content
	imagePath
	date

users	
PK	<u>uniqueId</u>
	email
	password
	username

dataservers	
PK	<u>uniqueId</u>
	rateCPU
	rateRAM
	rateDiskStorage
	date

Comment installer une dépendance ?

Pour l'installation d'une dépendance liée à Angular ou NodeJS, l'utilisation du gestionnaire de paquet de NodeJS, npm est recommandée. L'utilisation de cette commande se fait de deux façons :

```
npm install [-options] <name>
```

Ou par son alias :

```
npm i [-options] <name>
```

Par défaut, l'installation ne se fait que sur votre machine, les dépendances ne seront pas enregistrées dans le projet. Il est possible cependant de sauvegarder ces dépendances pour que le mainteneur puisse déployer correctement le système.

Pour sauvegarder cette dépendance pour la version de production :

```
npm install --save-prod <name>
```

Ou par son alias :

```
npm install -P <name>
```

Cependant pour des dépendances liées aux développements, aux tests, il est important de ne pas les sauvegarder en production, mais les garder seulement pour la version de développement :

```
npm install --save-dev <name>
```

Ou par son alias :

```
npm install -D <name>
```

Comment accéder aux données ?

En double-cliquant sur l'une des collections, il est possible d'avoir un premier aperçu de toutes nos entrées.

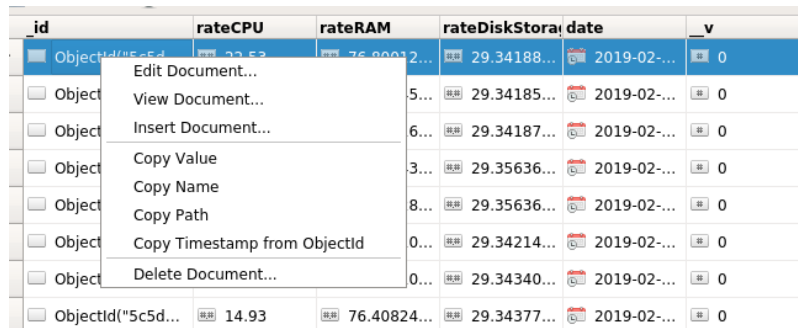
Key	Value	Type
(1) ObjectId("5c5db8cfbe15d01556f31ddb")	{ 6 fields }	Object
_id	ObjectId("5c5db8cfbe15d01556f31ddb")	ObjectId
rateCPU	22.53	Double
rateRAM	76.8001257317983	Double
rateDiskStorage	29.3418831691905	Double
date	2019-02-08 17:13:51.605Z	Date
_v	0	Int32
(2) ObjectId("5c5db8d0be15d01556f31ddc")	{ 6 fields }	Object
(3) ObjectId("5c5db8d1be15d01556f31ddd")	{ 6 fields }	Object
(4) ObjectId("5c5db8d2be15d01556f31dde")	{ 6 fields }	Object
(5) ObjectId("5c5db8d3be15d01556f31ddf")	{ 6 fields }	Object
(6) ObjectId("5c5db8d4be15d01556f31de0")	{ 6 fields }	Object

En haut à droite de cette fenêtre, il est possible d'avoir plus d'informations sur les différentes entrées et ainsi obtenir un affichage plus classique pour les utilisateurs de base de données SQL, via l'icône :

	_id	rateCPU	rateRAM	rateDiskStorage	date	_v
1	ObjectId("5c5d...	22.53	76.80012...	29.34188...	2019-02-...	0
2	ObjectId("5c5d...	13.9	76.78745...	29.34185...	2019-02-...	0
3	ObjectId("5c5d...	23.75	77.12216...	29.34187...	2019-02-...	0

Comment éditer les données ?

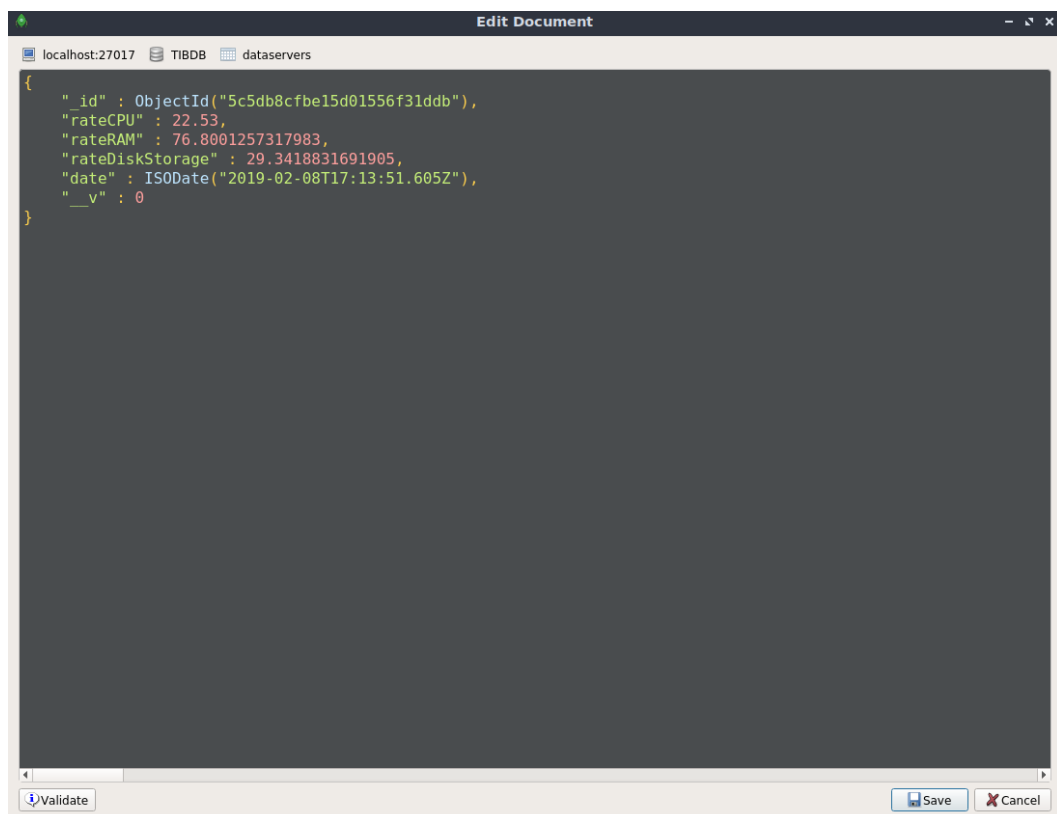
En étant sur l'une des différentes vues d'une collection, il est possible de modifier une entrée, pour cela il faut faire un clic-droit sur l'une des entrées puis sélectionner l'option « Edit Document » (via l'option « Insert Document », il est possible d'ajouter une entrée)



The screenshot shows a table with columns: **_id**, **rateCPU**, **rateRAM**, **rateDiskStorage**, **date**, and **_v**. A right-click context menu is open over the first row, showing options: Edit Document..., View Document..., Insert Document..., Copy Value, Copy Name, Copy Path, Copy Timestamp from Objectid, and Delete Document... The table contains several rows of data with various numerical and date values.

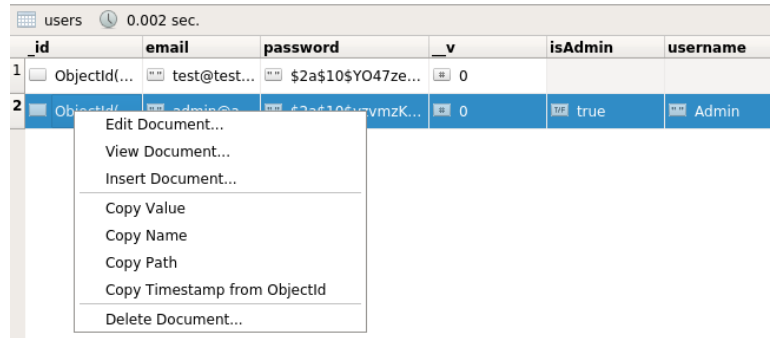
_id	rateCPU	rateRAM	rateDiskStorage	date	_v
ObjectId("5c5d...")	22.53	76.80012...	29.34188...	2019-02-...	0
Object...			29.34185...	2019-02-...	0
Object...			29.34187...	2019-02-...	0
Object...			29.35636...	2019-02-...	0
Object...			29.35636...	2019-02-...	0
Object...			29.34214...	2019-02-...	0
Object...			29.34340...	2019-02-...	0
ObjectId("5c5d...")	14.93	76.40824...	29.34377...	2019-02-...	0

Après le clic sur l'option « Edit Document », on obtient cette vue et ainsi on peut modifier toutes les valeurs de l'entrée puis sauvegarder ou annuler nos modifications via les boutons en bas à droite de cette fenêtre :



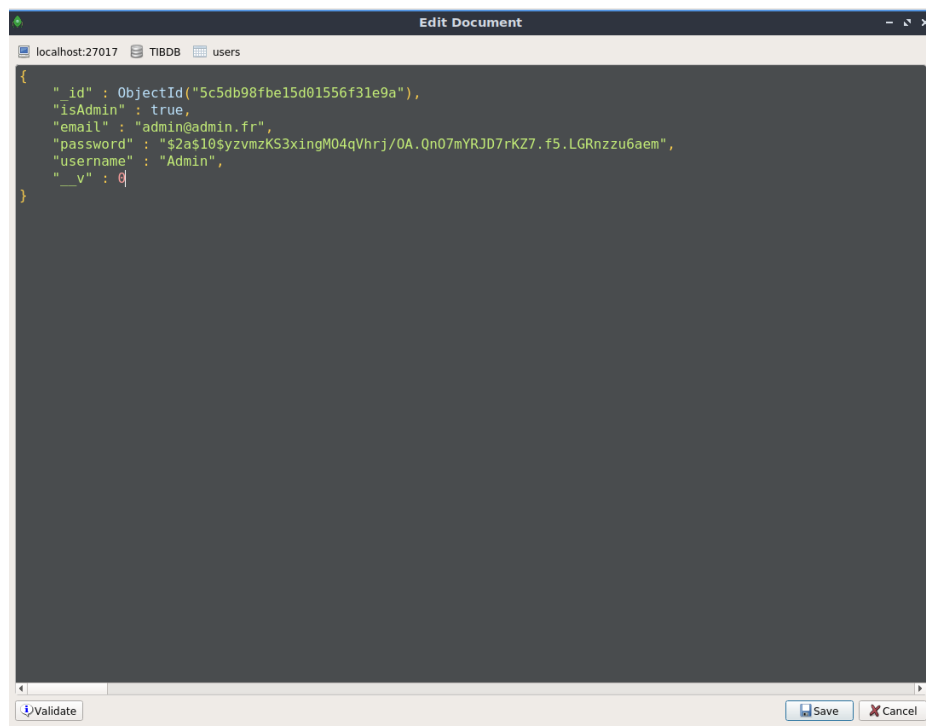
Comment passer un utilisateur en administrateur ?

Actuellement, le seul moyen de passer un utilisateur en administrateur est de modifier (« Edit Document ») l'entrée correspondant à l'utilisateur souhaitée, dans la collection « users » présent dans la base de données lié à l'application Web.



	id	email	password	v	isAdmin	username
1	ObjectId(...)	test@test...	\$2a\$10\$YO47ze...	0		
2	ObjectId(...)	admin@...	\$2a\$10\$yvmzK...	0	true	Admin

Puis via la fenêtre d'édition, il est possible de passer la valeur « isAdmin » à true, afin que l'utilisateur possède les droits administrateurs sur l'application Web et donc puisse accéder aux différentes actions réservées aux administrateurs.



L'inverse est possible, pour repasser un administrateur en tant que simple utilisateur, en réalisant la même manipulation mais remplacer la valeur « isAdmin » par false.

Comment éditer les scripts d'une application ?

Chaque script pour une application sont stockés dans le dossier du nom de l'application présent dans le dossier « MEANStack\AngularPrj\tib\backend\scripts » en partant de la racine du projet Git.

Plusieurs scripts sont présents, un pour démarrer l'application, un pour mettre à jour l'application et un 3^e qui n'est pas encore mis en place actuellement celui pour arrêter correctement l'application. Tous les scripts sont modifiables et exécutables comme n'importe quel script Shell.

Comment déployer l'application Web ?

Une fois l'application compiler par le développeur, il est simple de déployer l'application Web via un seul et même serveur qui va gérer la partie Front-End & Back-End. Pour ce faire, il faut récupérer le contenu du dossier « backend » du projet Git, présent dans le dossier « MEANStack\AngularPrj\tib » puis lancer le serveur via la commande :

```
> npm run start:server
```

On peut retrouver cette commande dans le fichier de configuration « package.json ».